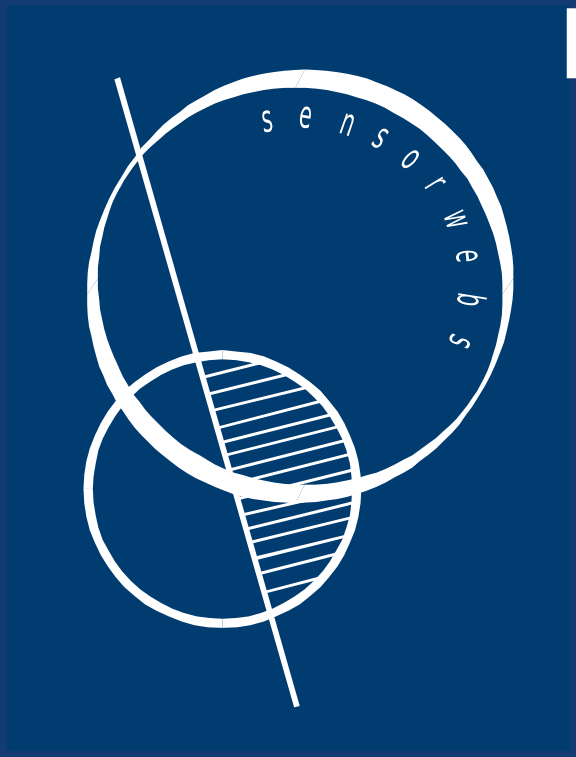


Distributed network signal processing

Sensorwebs group

http://basics.eecs.berkeley.edu/sensorwebs

Kannan Ramchandran

(Pister, Sastry,Anantharam,Jordan,Malik)

Electrical Engineering and Computer Science

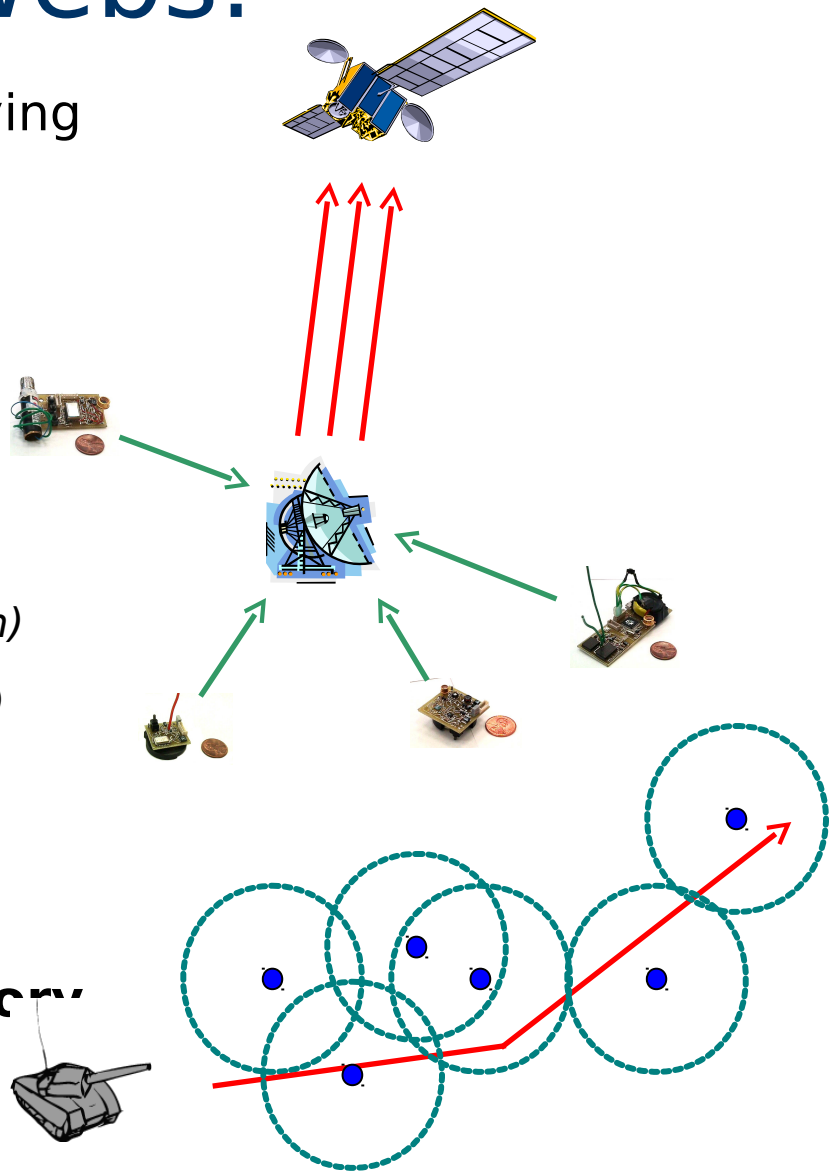
kannanr@eecs.berkeley.edu

http://www.eecs.berkeley.edu/~akannanr

University of California at

DARPA Sensorwebs:

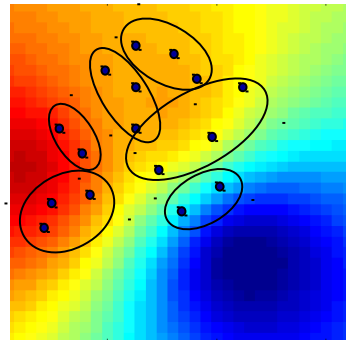
- Creation of a fundamental unifying framework for real-time distributed/ decentralized information processing with applications to Sensor Webs, consisting of:
  - **MEMS** *(Pister)*
  - **Distributed SP** *(Ramchandran)*
  - **Distributed Control** *(Sastry)*
  - **"Real-time" Information Theory** *(Anantharam)*
  - **Distributed Learning Theory** *(Jordan)*



Dense low-power sensor network attributes

- Dense clustering of sensors/ embedded devices
- Highly correlated but spatially distributed data
- Limited system resources: energy, bandwidth
- Unreliable system components
- Wireless medium: dynamic SNR/ interference
- End-goal is key: tracking, detection, inference

Disaster Management



Signal processing & comm. system challenges:

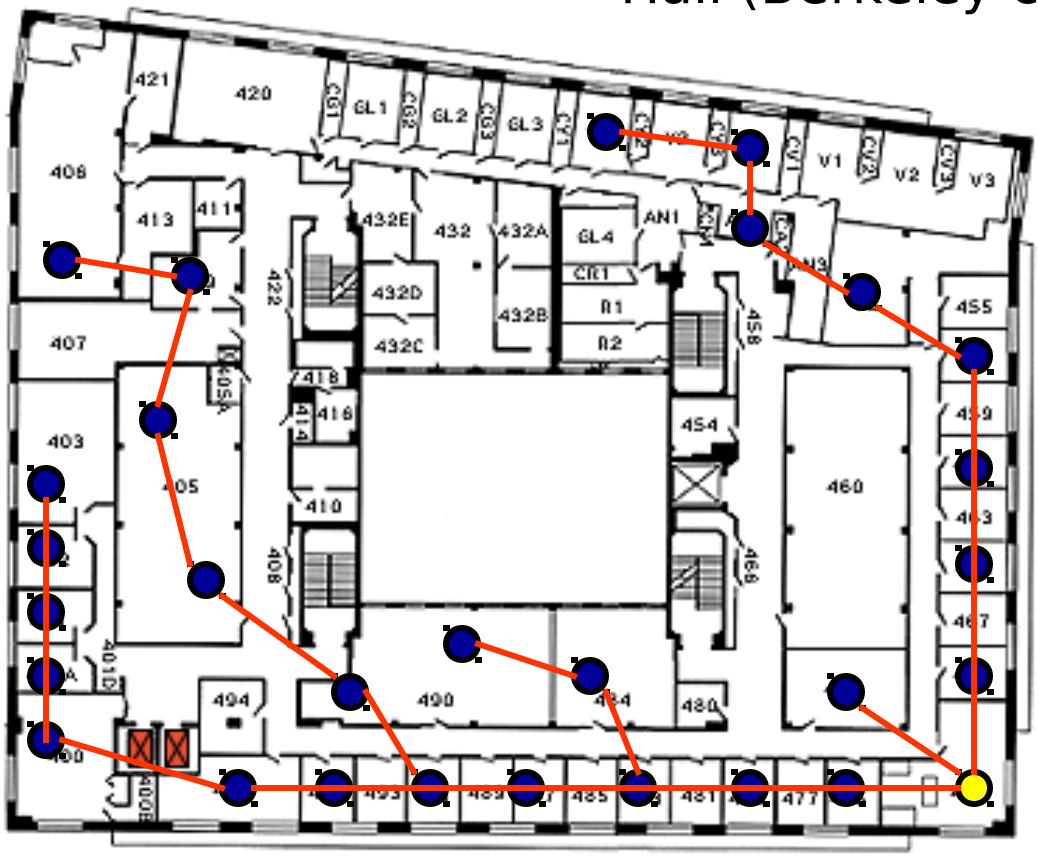
- Distributed & scalable multi-terminal architectures for:
  - **coding, clustering, tracking, estimation, detection;**
- Distributed sensor fusion based on statistical sensor data models;
- Integration of layers in network stack:
  - **joint source-network coding**
  - **joint coding/routing**
- Reliability through diversity in representation & transmission
- Energy optimization: computation vs. transmission cost:
  - **~100 nJ/bit vs. 1 pJ/inst. (HW) & 1 nJ/inst (SW)**

Roadmap

- **Distributed Compression**: basics, new results
- **Networking aspects:** packet aggregation
- **Reliability through diversity:** multiple descriptions
  - **Distributed multimedia streaming:** robust, scalable architecture

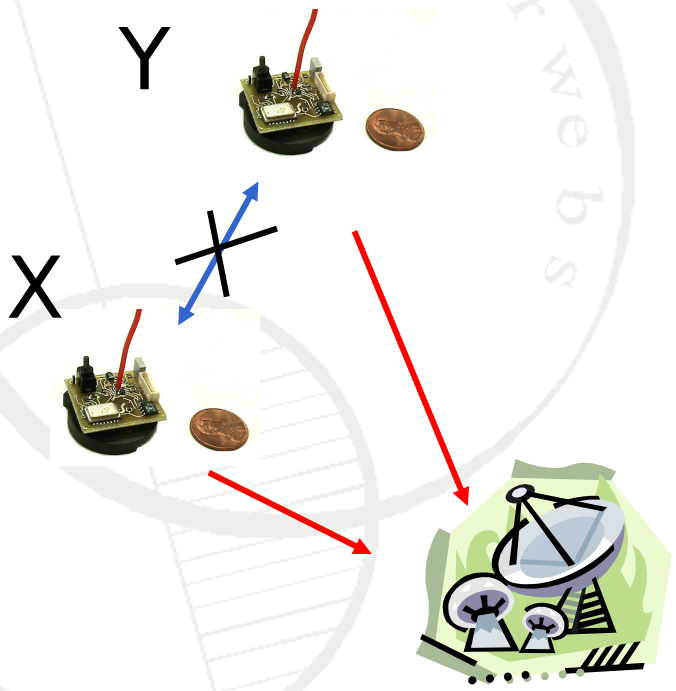
Real-world scenario: Blackouts project

- Near Real-time room condition monitoring using sensor motes in Cory Hall (Berkeley campus): data goes online.
  - All sensors periodically route readings to central (yellow) node.
  - Strong multiplication of redundancy due to topology.



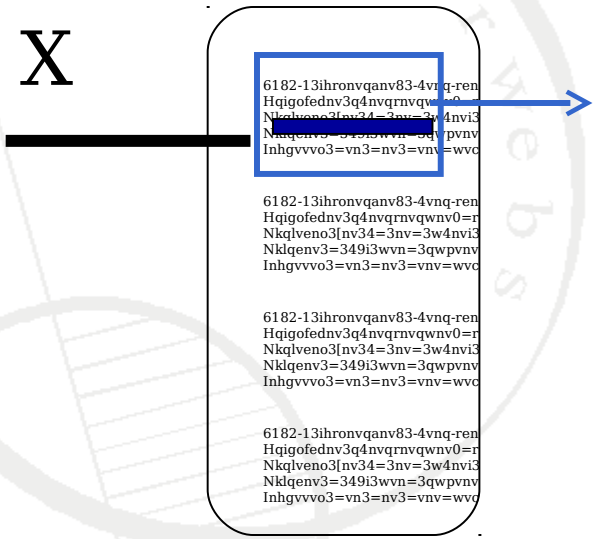
- *http://blackouts.eecs.berkeley.edu*

Distributed compression: basic ideas



- Suppose X, Y correlated
- Y available at decoder but not at encoder
- How to compress X close to H(X|Y)?
- Key idea: discount I(X;Y). H(X|Y) = H(X) – I(X;Y)

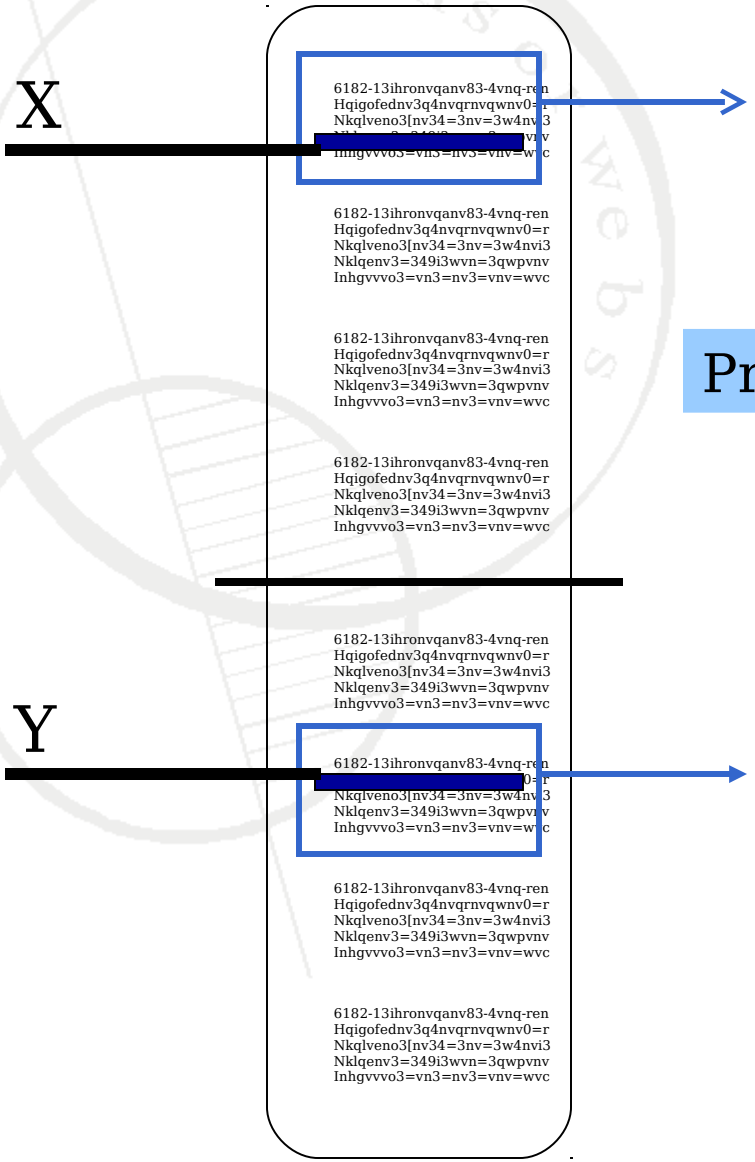
Information-theory: binning argument



Slepian-Wolf ('72)

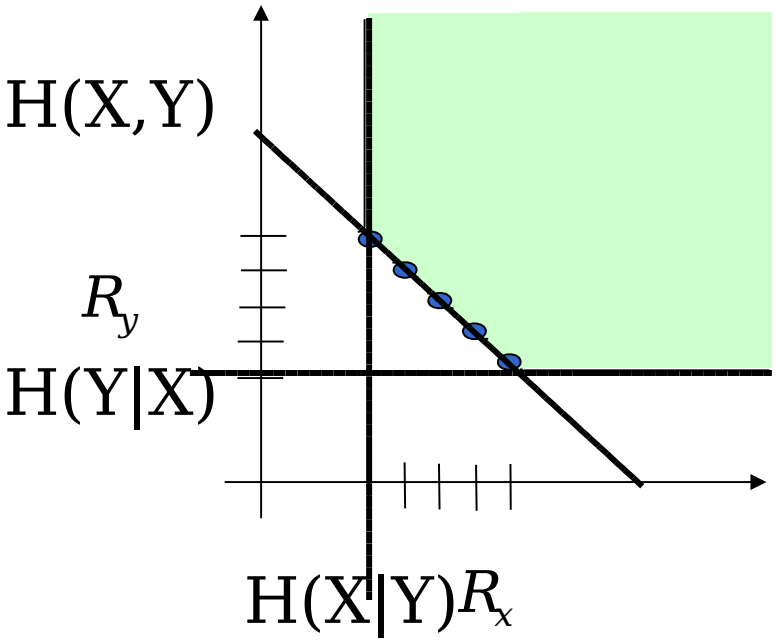
- Make a main codebook of all typical sequences. $2^{nH(X)}$ and $2^{nH(Y)}$ elements.
- Partition into $2^{nH(X|Y)}$.
- When observe X^n, transmit index of bin it belongs to
- Decoder finds member of bin that is jointly typical with Y^n.
- Can extend to "symmetric cases"

Symmetric case: joint binning



- Rate limited by:
  $R_x \geq H(X|Y)$
  $R_Y \geq H(Y|X)$
  $R_x + R_Y \geq H(X,Y)$

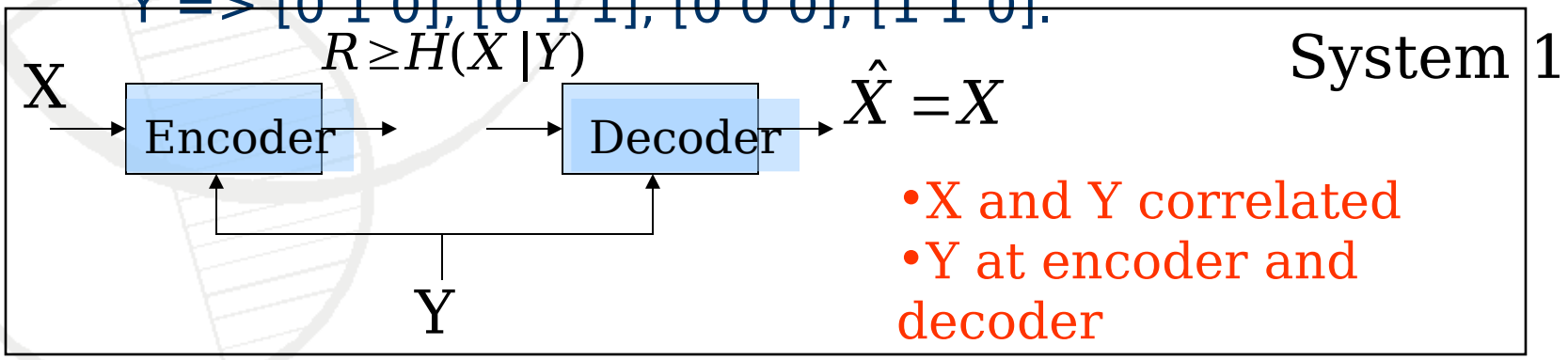
Product of bin siz $2^{nI(X,Y)}$



Simple binary example

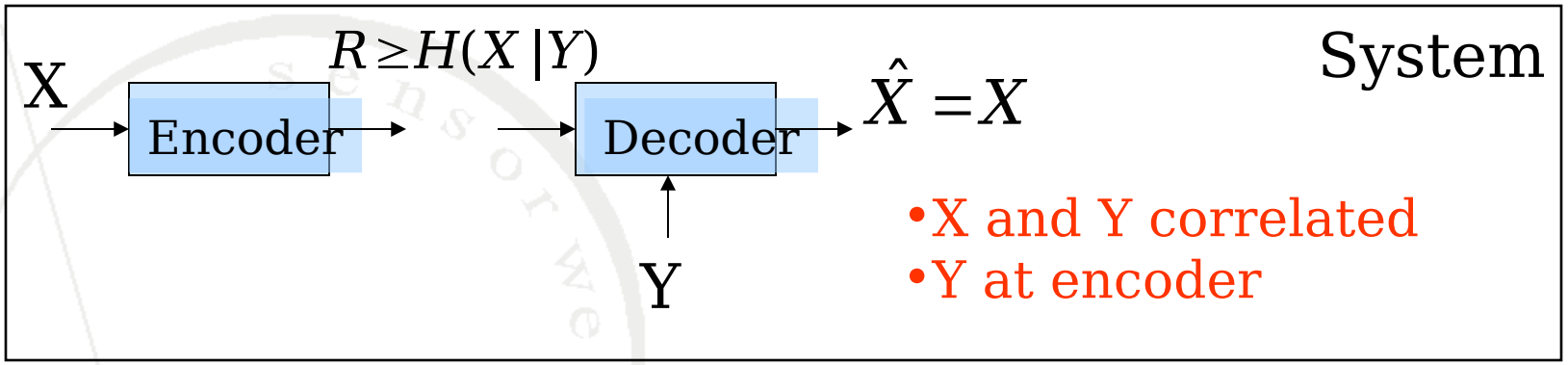
- X and Y => length-3 binary data (equally likely),
- Correlation: Hamming distance between X and Y is at most 1.

Example: When X=[0 1 0],
Y => [0 1 0], [0 1 1], [0 0 0], [1 1 0].



$$X+Y=\begin{cases}0\ 0\ 0\\0\ 0\ 1\\0\ 1\ 0\\1\ 0\ 0\end{cases}$$

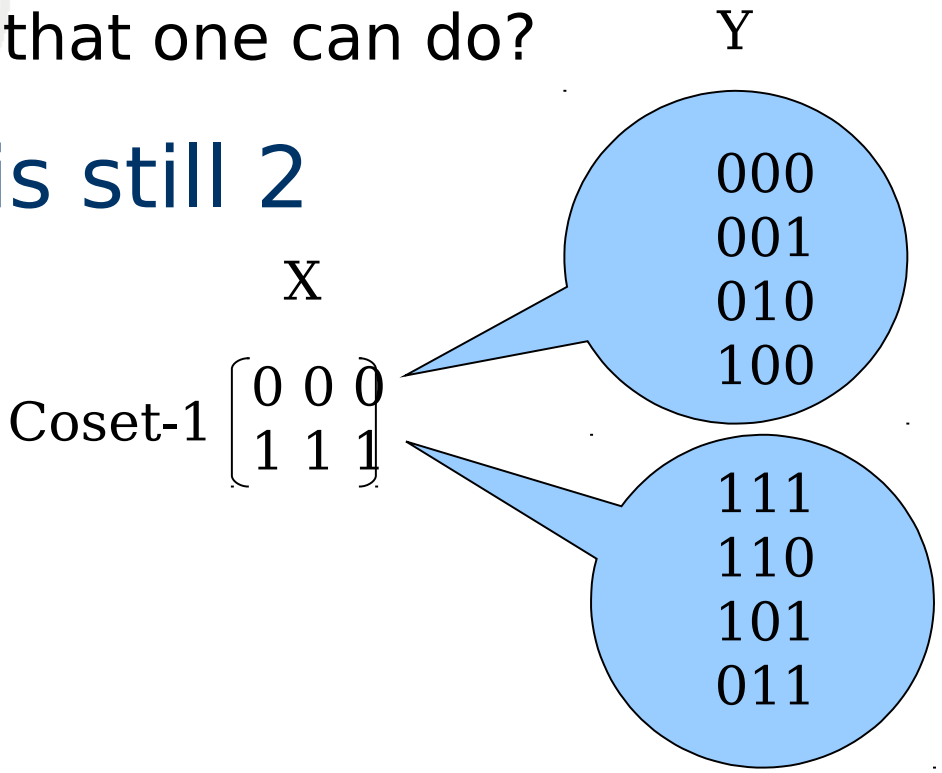
Need 2 bits to index this.



- What is the best that one can do?

- The answer is still 2 bits!

How?



Coset-1 $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Coset-2 $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Coset-3 $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

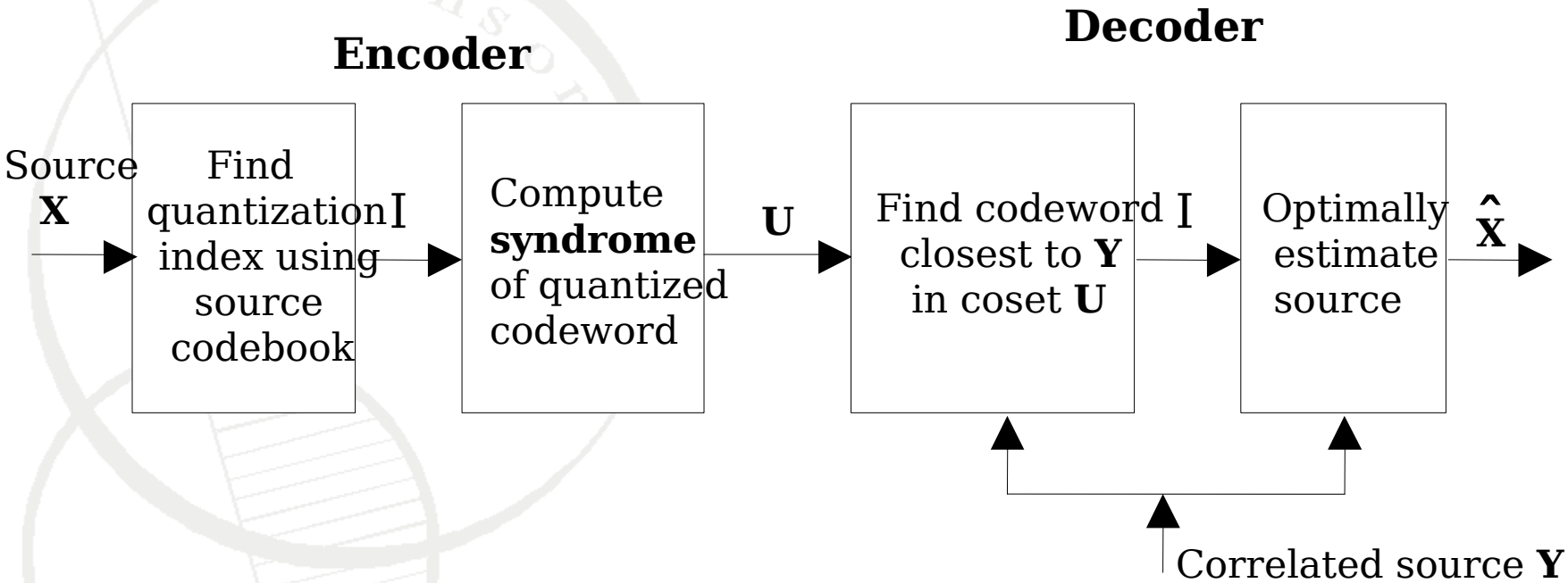
Coset-4 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

- Encoder -> index of the coset containing X.
- Decoder reconstructs X in given coset.

Note:

- Coset-1 -> repetition code.
- Each coset -> unique “syndrome”
- DIstributed Source Coding Using Syndromes (DISCUS)

General block diagram of DISCUS

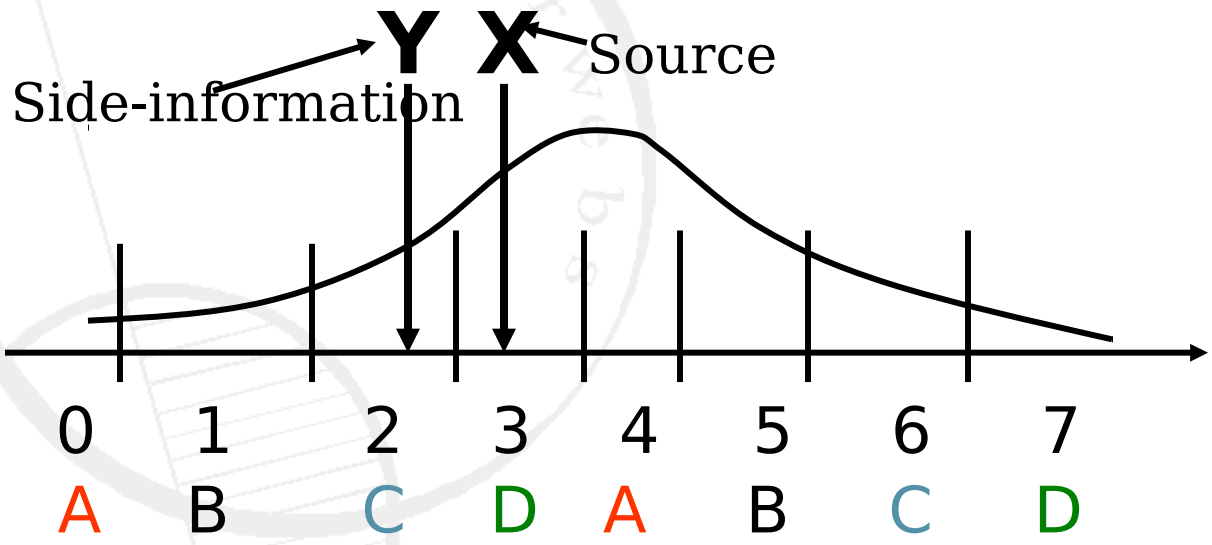


DISCUS: a constructive approach to distr. compression

- Intricate interplay between source coding, channel coding and estimation theory: can leverage latest advances in all areas.
- **7-15 dB** gains in reconstruction SNR over theoretically optimal strategies that ignore correlation for typical correlated sources.

Continuous case: practical quantizer design issues

- Consider the following coset example: 8-level scalar quantizer

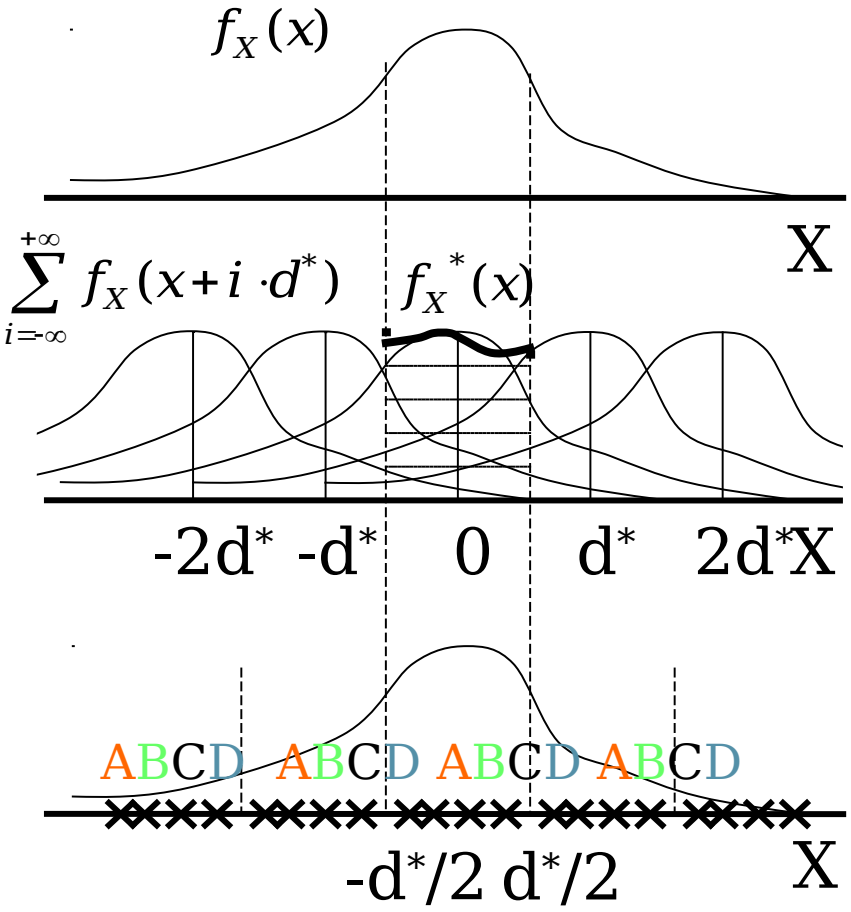


- Difference at most 1 cell.
- Send only index of “coset”: A,B,C,D
- Decoder decides which member of coset is the correct answer

We have compressed from 3 bits to 2 bits

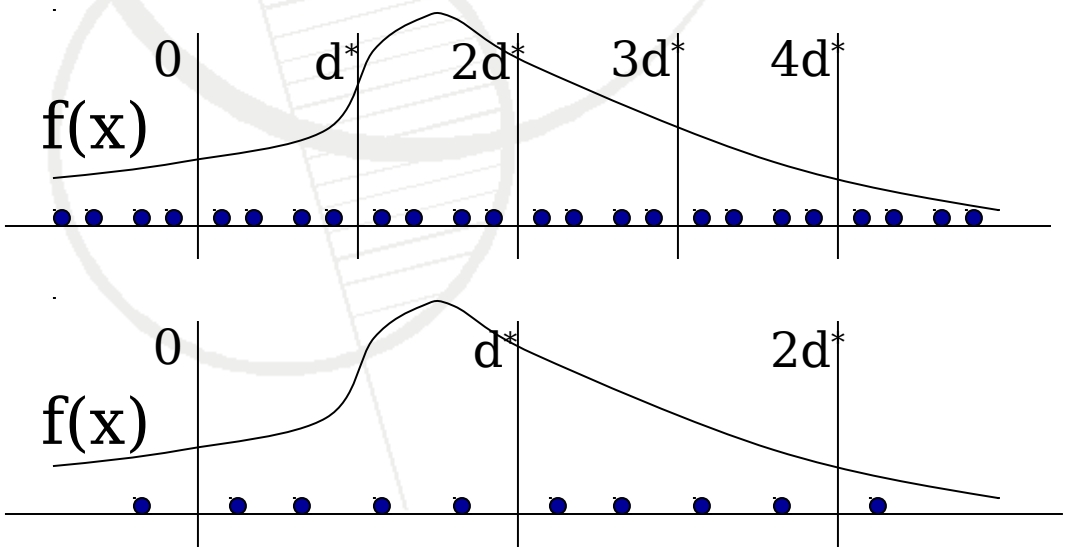
Optimizing quantizer and rate

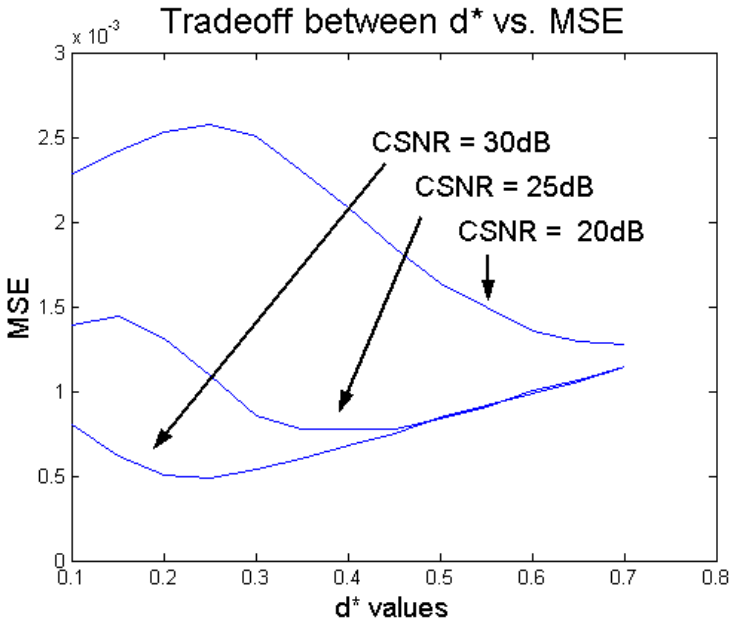
- Important note: decoder can't differentiate bet. x and x+d* (ABCD)
- Therefore: must combine statistics of members of bins
- Use **PDF periodization**: repeat PDFs using parameter d*.
- Design using **f'$_x$(x)**



Caveats: choice of d*

- If too small: high coset error
- If too large: high quantization error

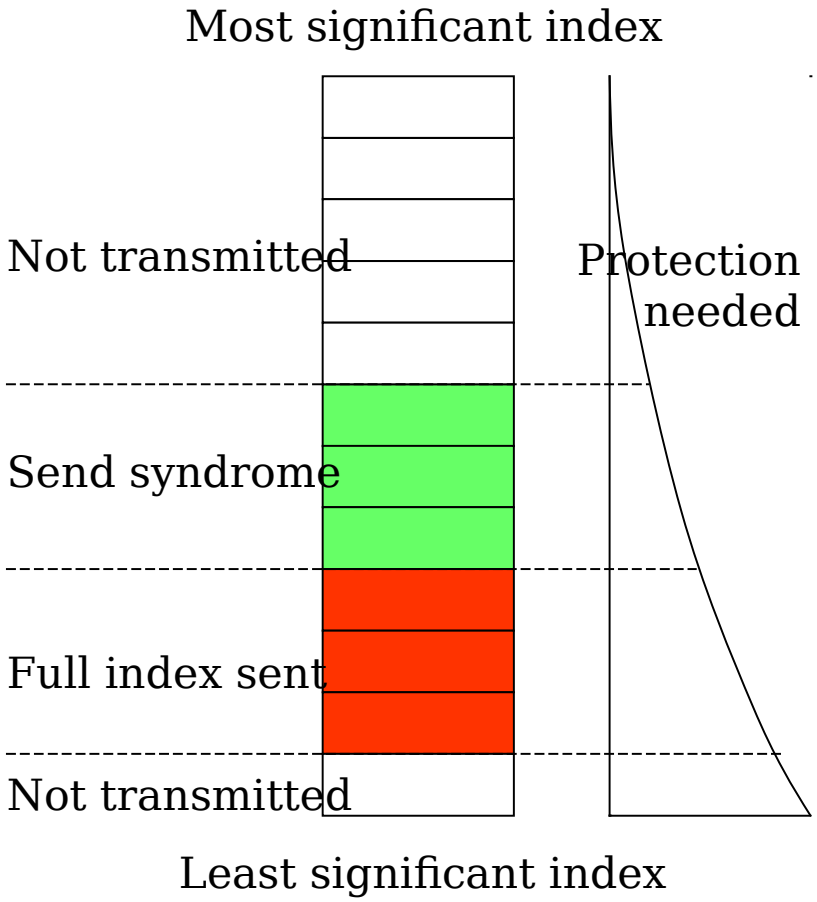




Kusuma & Ramchandran ‘01

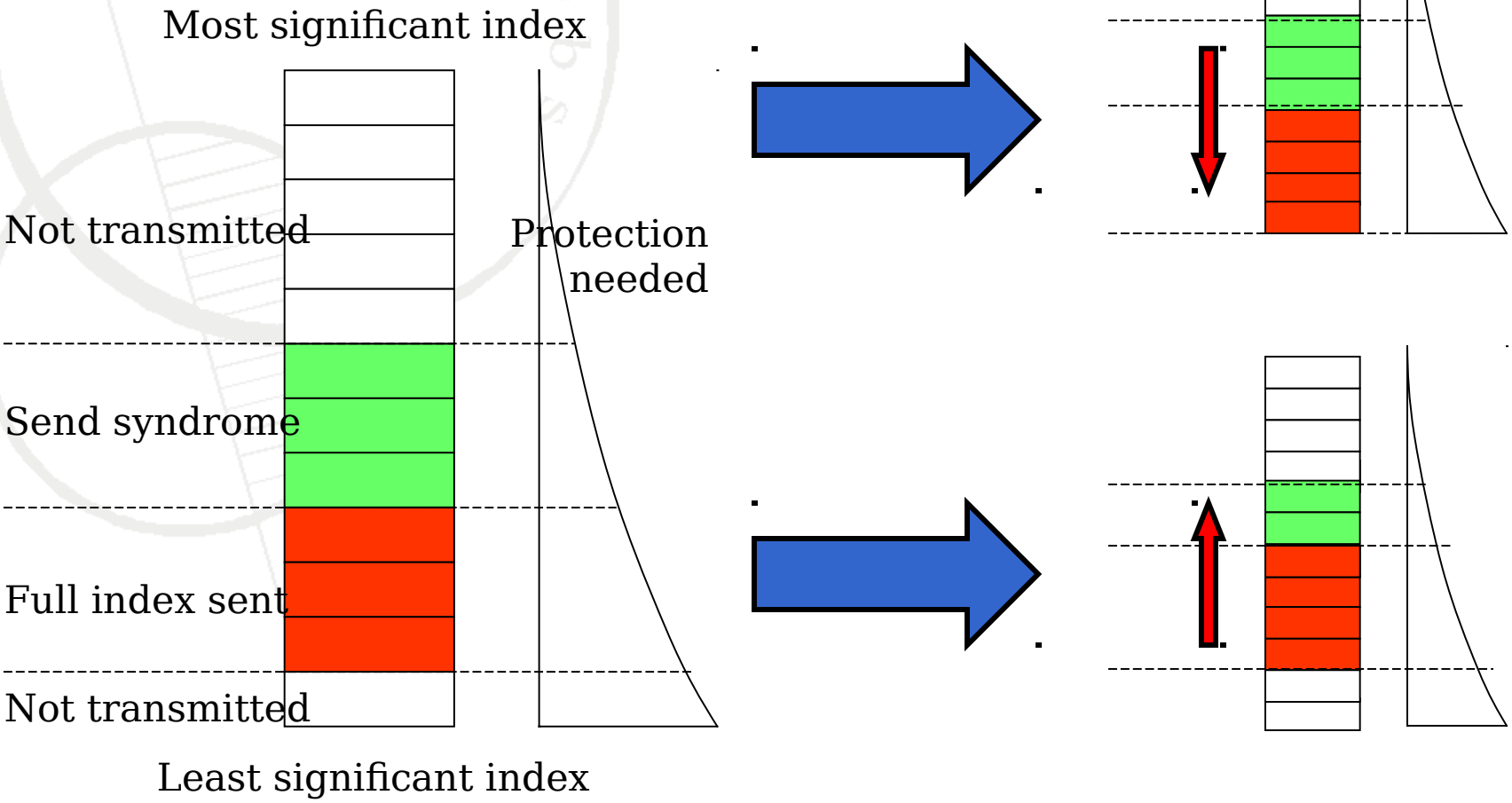
Dynamic bit allocation

- Consider iterative method: assign one bit at a time
- Can either:
  improve quantization
  improve code performance
- Iteratively assign using rules of thumb.
- Multiple levels of protection



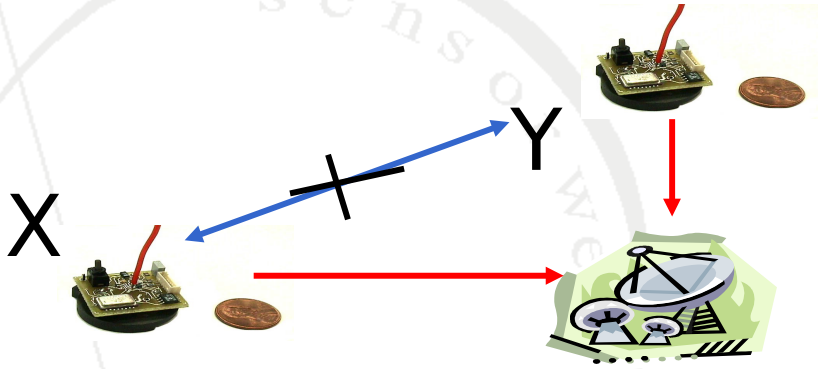
For example

- Increase quantization (need more protection too!) OR
- Increase code performance



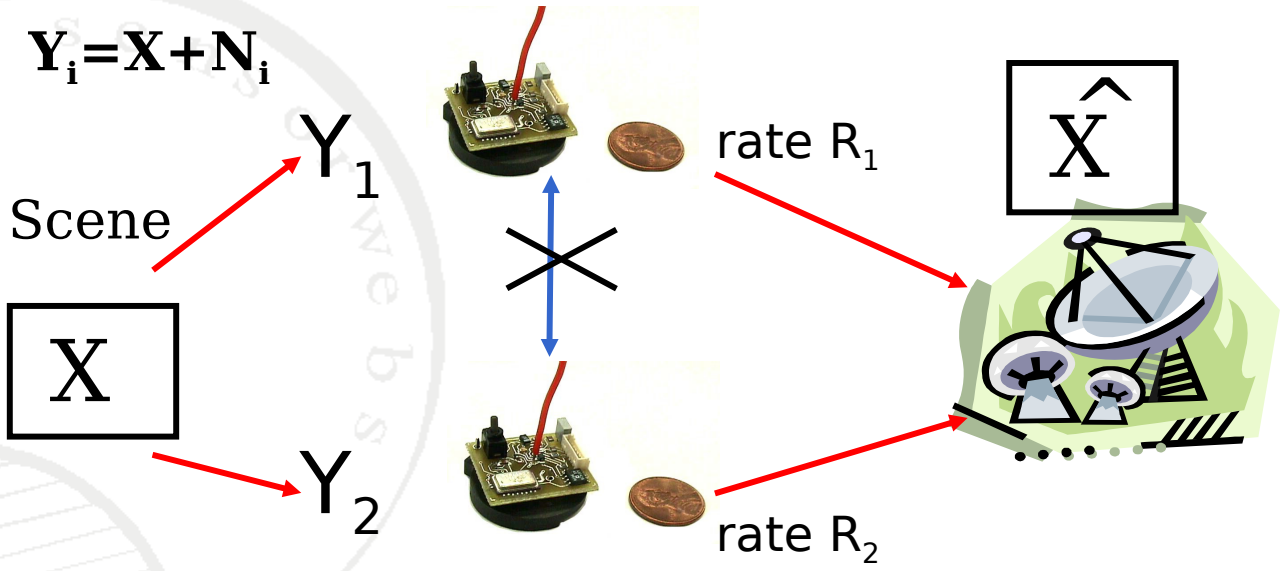
New results: distributed lossy compression:

(Pradhan & Ramchandran ’01)



- Suppose X, Y correlated as X=Y+N
- **Wyner-Ziv (’78):** No theoretical performance loss due to distributed processing (no X-Y communication) if X and Y are jointly Gaussian.
- New results (Pradhan, Chou & Ramchandran ’01)::
  - No performance loss due to distributed processing for ***arbitrary*** X, Y if N is Gaussian.
  - Fundamental **duality** between distributed coding and data- hiding (encoder/decoder functions can be swapped!)

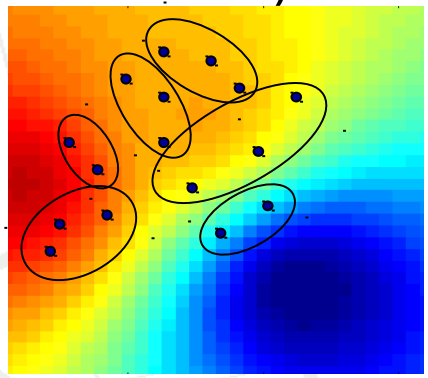
Distributed sensor fusion under bandwidth constraints:



- Suboptimal to form $E(X|Y_i)$ as in single-sensor case
- Optimal distributed strategy for Gaussian case:
  - Compress Y_i's without estimating X individually.
  - Exploit correlation structure to reduce transmitted bit rate.
  - DISCUS ➔ multisensor fusion under BW constraints.

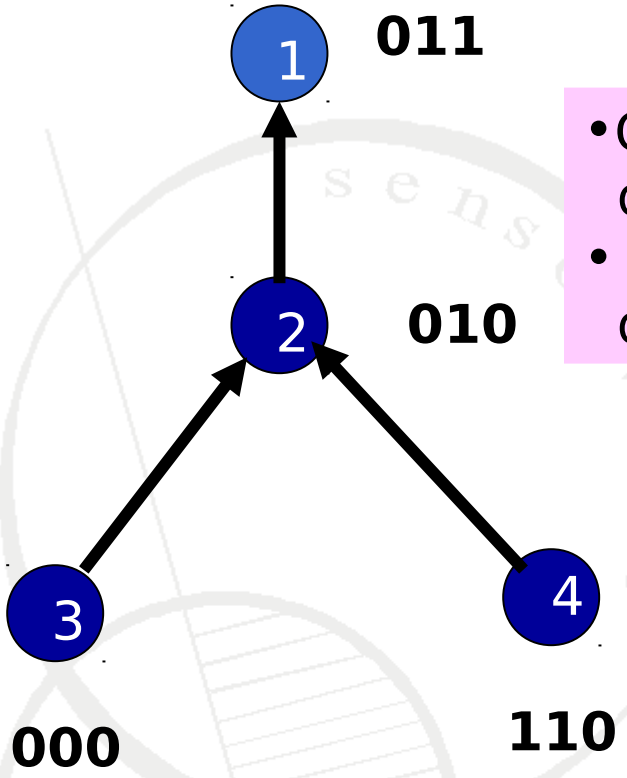
Enabling DISCUS for sensor networks

- Use clustering to enable network deployment (hierarchies)

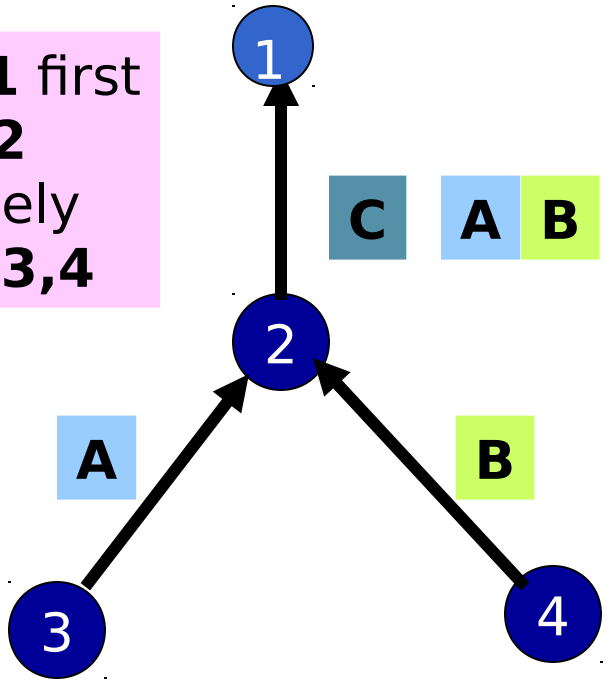


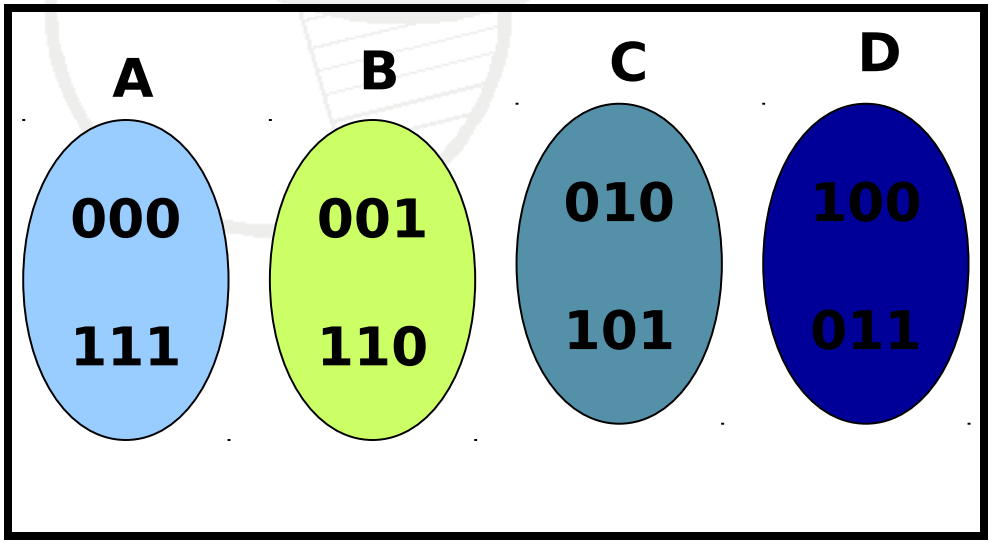
  - Elect a “cluster leader” – can swap periodically.
  - Localization increases robustness to changing correlation structure (everything is relative to leader).

- Learn correlation structure (training based or dynamically) and optimize quantizer and code:
  - Good news: no need for motes to be aware of clustering



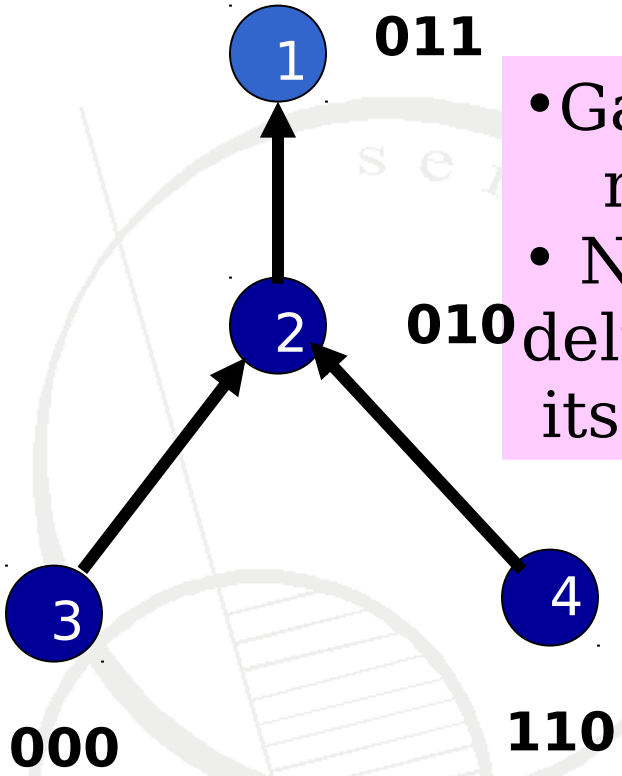
- Gateway node **1** first decodes node **2**
- It then recursively decodes nodes **3,4**



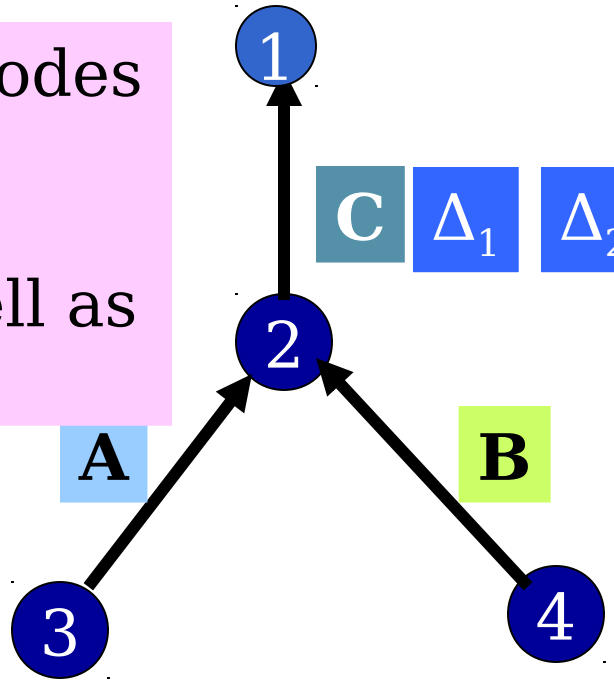


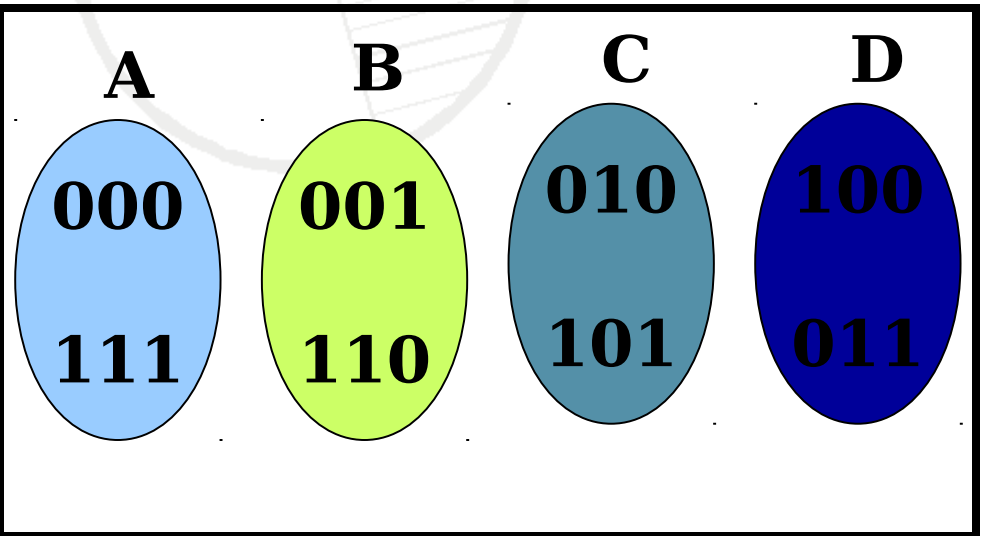
If each link ~ 1 m, network doe
15 bit-meter work **w/o DISCU**

With DISCUS, network
does only **10 bit-meter** work.



- Gateway node **2** decodes nodes **3,4**
- Node **2** sends the deltas w.r.t. **3,4** as well as its own syndrome



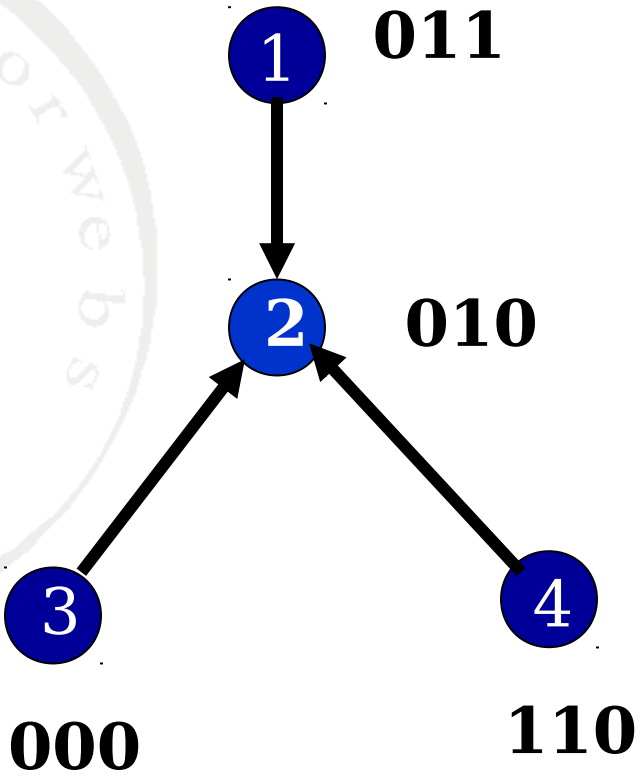


If each link ~ 1 m, network
15 bit-meter work **w/o DI**

With DISCUS, network
does only **10 bit-meter** wo

Load-balance in computatio

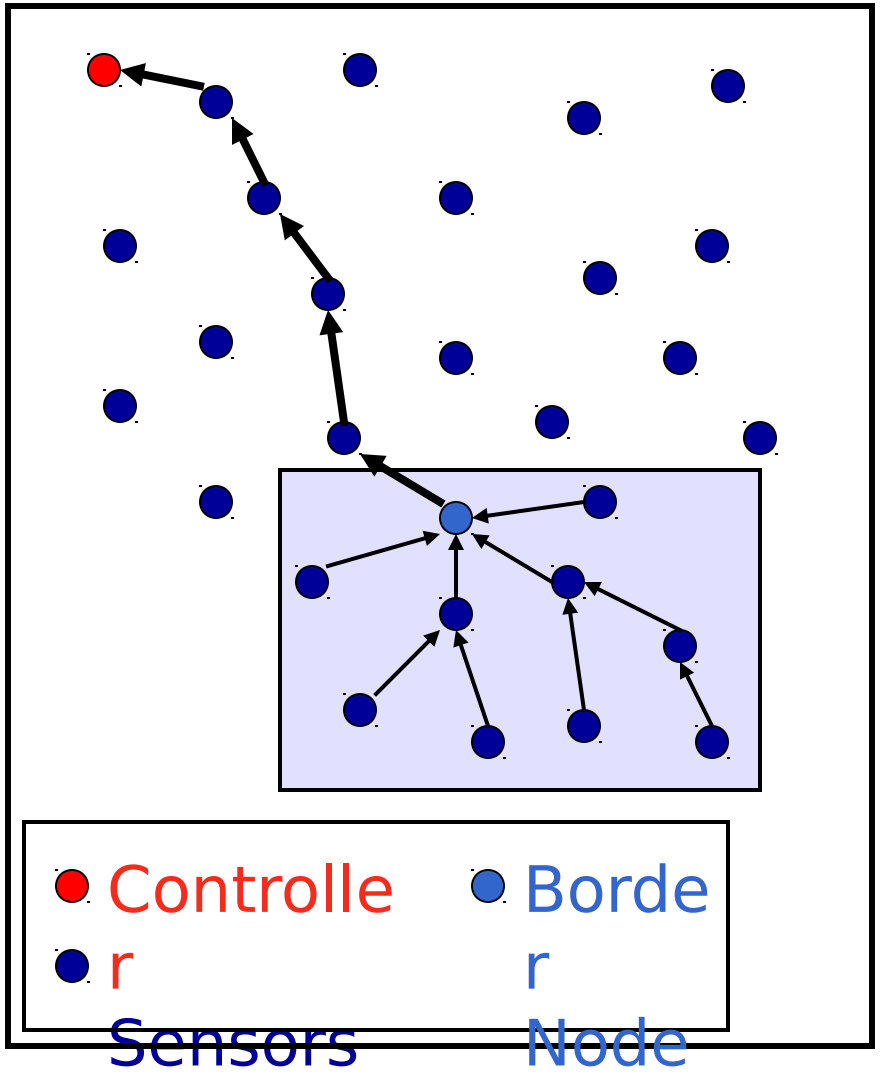
Where should aggregation be done?



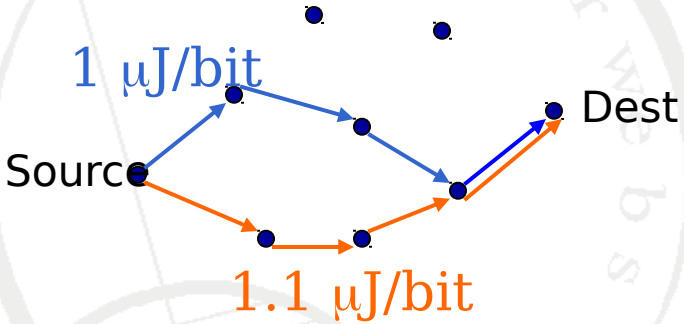
Node **2** collects all the data: nodes **1,3,4** send 2-bit syn
Total work done by network down to **6 bit-meters**!

Network deployment: aggregation (Picoradio project BWRC)

- Sensor nodes from region of interest (ROI) reply to query
- ROI sends out single aggregate packet



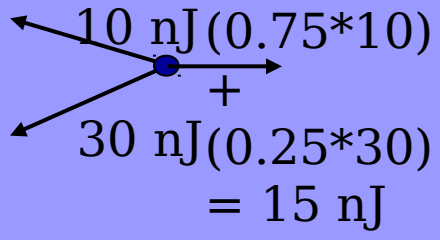
Integration into routing protocols

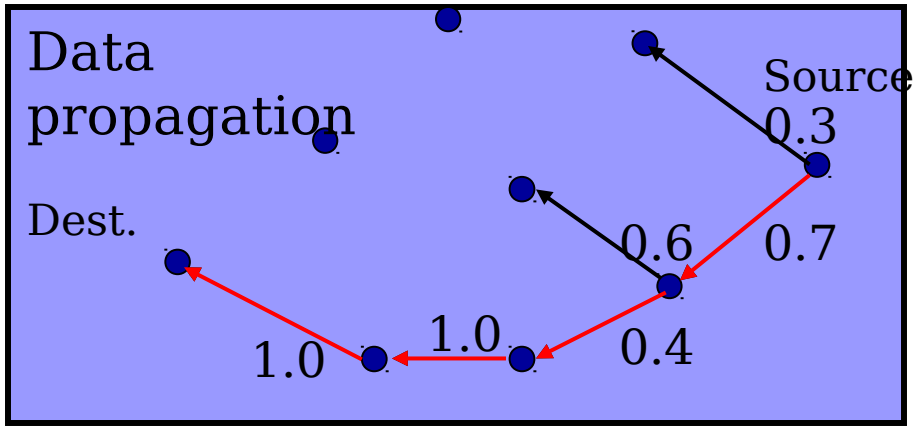


- Traditional way: find the best route and
  use it always
- Probabilistic path selection is superior
- Can incorporate data correlation structure
  into path weights

Local Rule

$p_1 = 0.75$
$p_2 = 0.25$



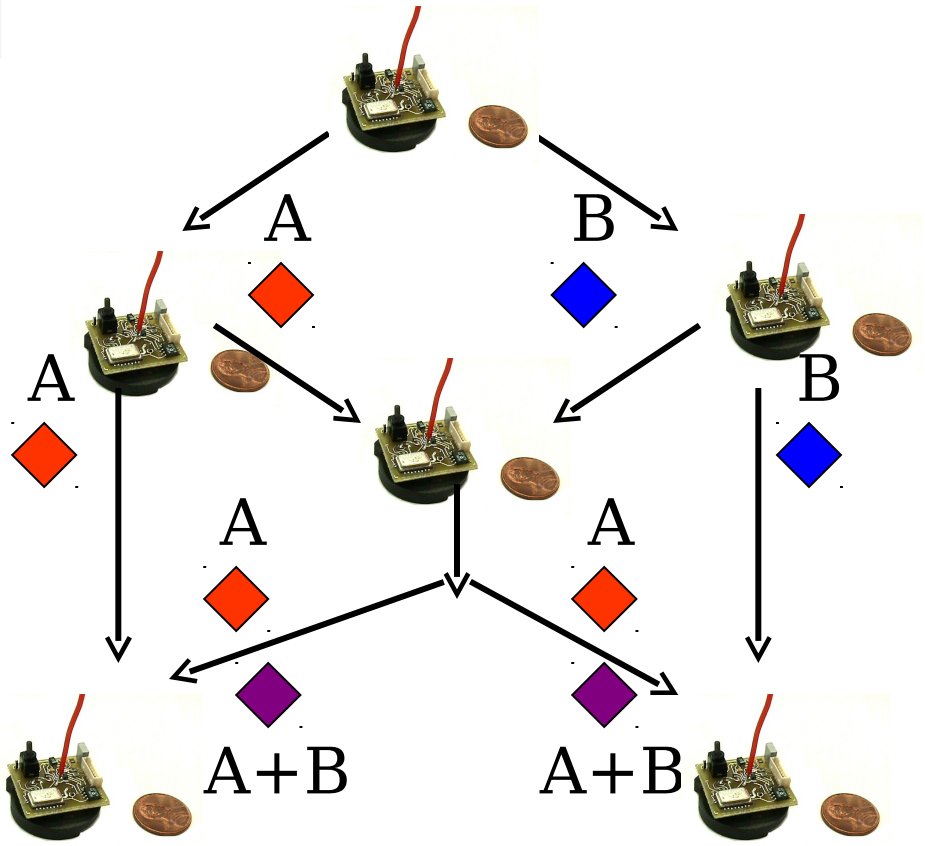


Network Coding: the case for “smart motes”

- “Store and forward” philosophy of current packet routers can be
- “Smart motes” can significantly decrease system energy require

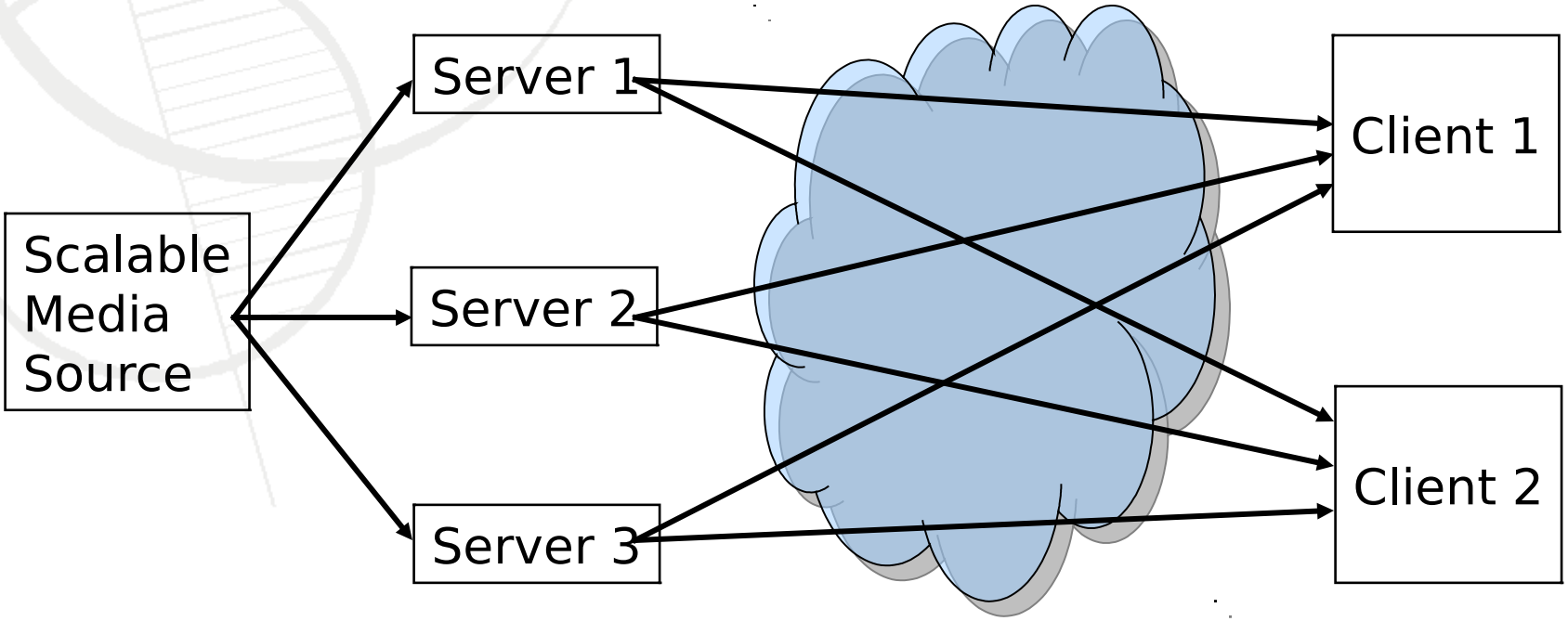
Information A

Information B

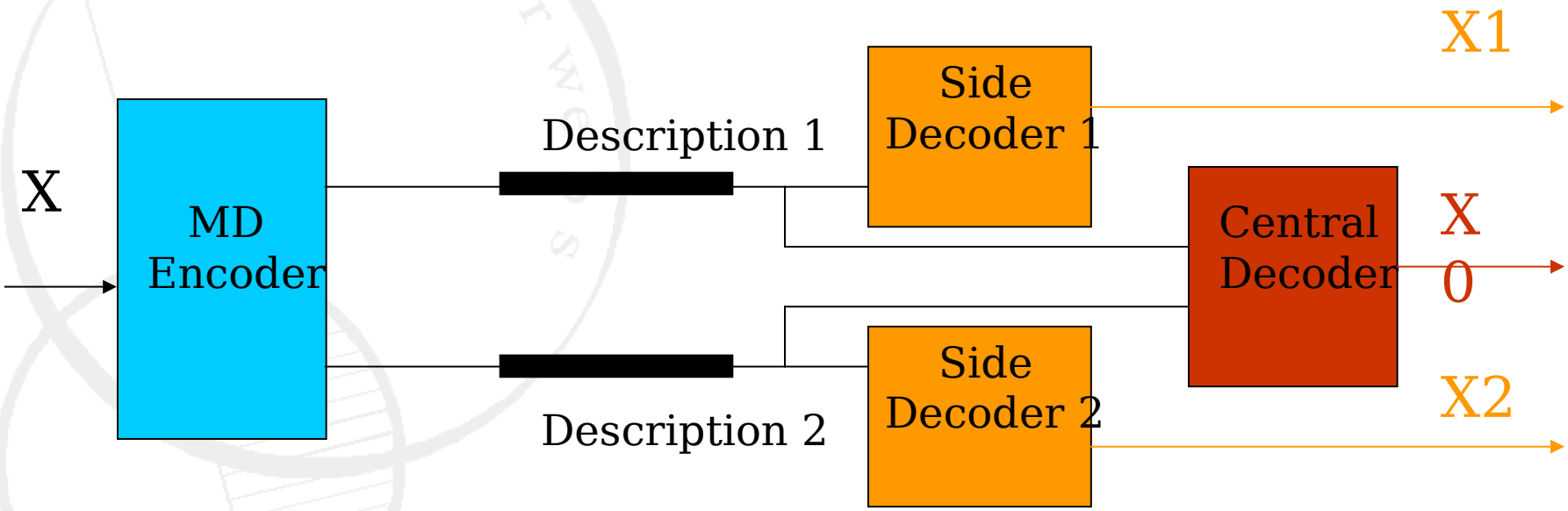


Distributed Media Streaming from Multiple Servers: new paradigm

- Client is served by multiple servers
- Advantages:
  - Robust to link/server failure (useful in battlefield!)
  - Load balancing



Robust transmission: the Multiple Descriptions Problem



X0 < X1, X2

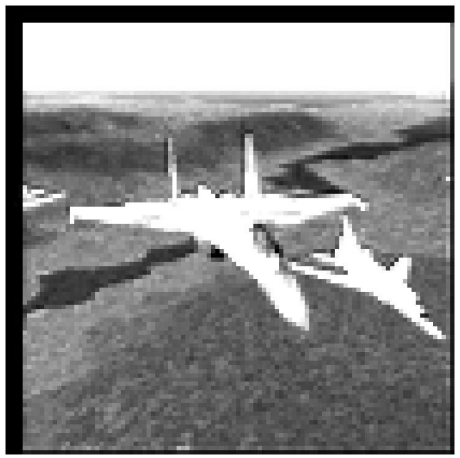
Distortion X1 = X2

- **Multiple levels of quality delivered to the destinat**
  levels for the N-channel case)

Emerging Multimedia Compression Standards

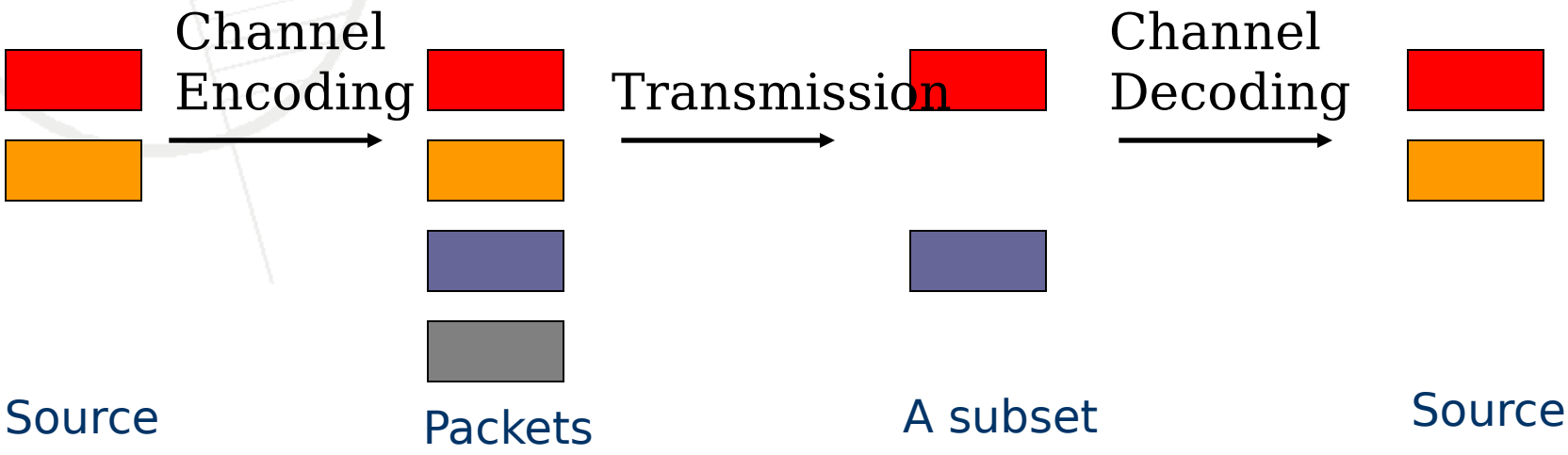
- Multi-resolution (MR) source coding e.g, JPEG-2000 (wavelets), MPEG-4.

- Bit stream arranged in importance layers (progressive)

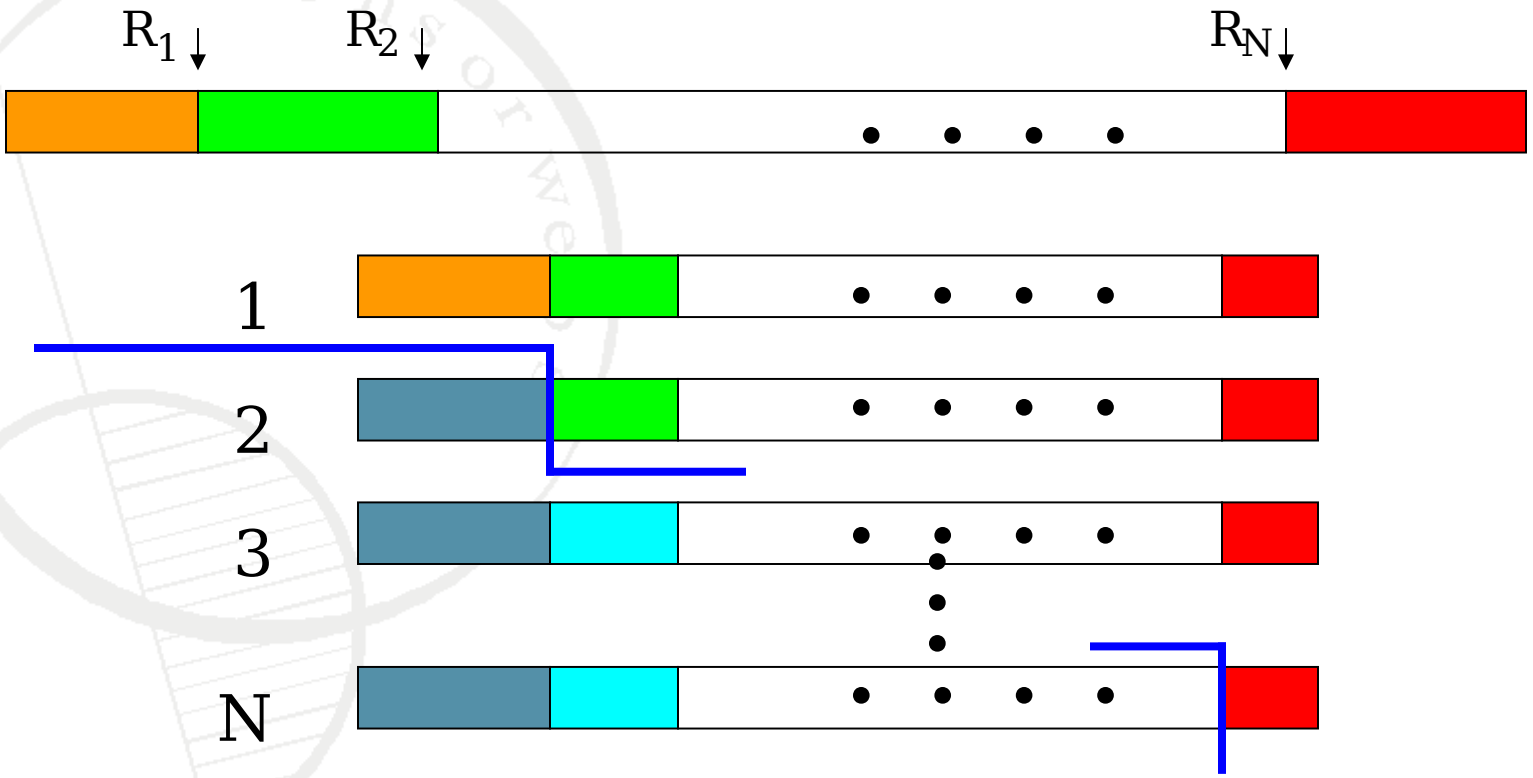


A Review of Erasure Codes

- Erasure Codes **(n, k, d)** : recovery from reception of partial data.
- n = block length, k= log (# of code words), correct (d-1) erasures
- **(n,k)** Maximum Distance Separable (MDS) Codes: **d = n – k + 1**
- MDS => *any* **k** channel symbols => **k** source symbols.



Robust Source Coding

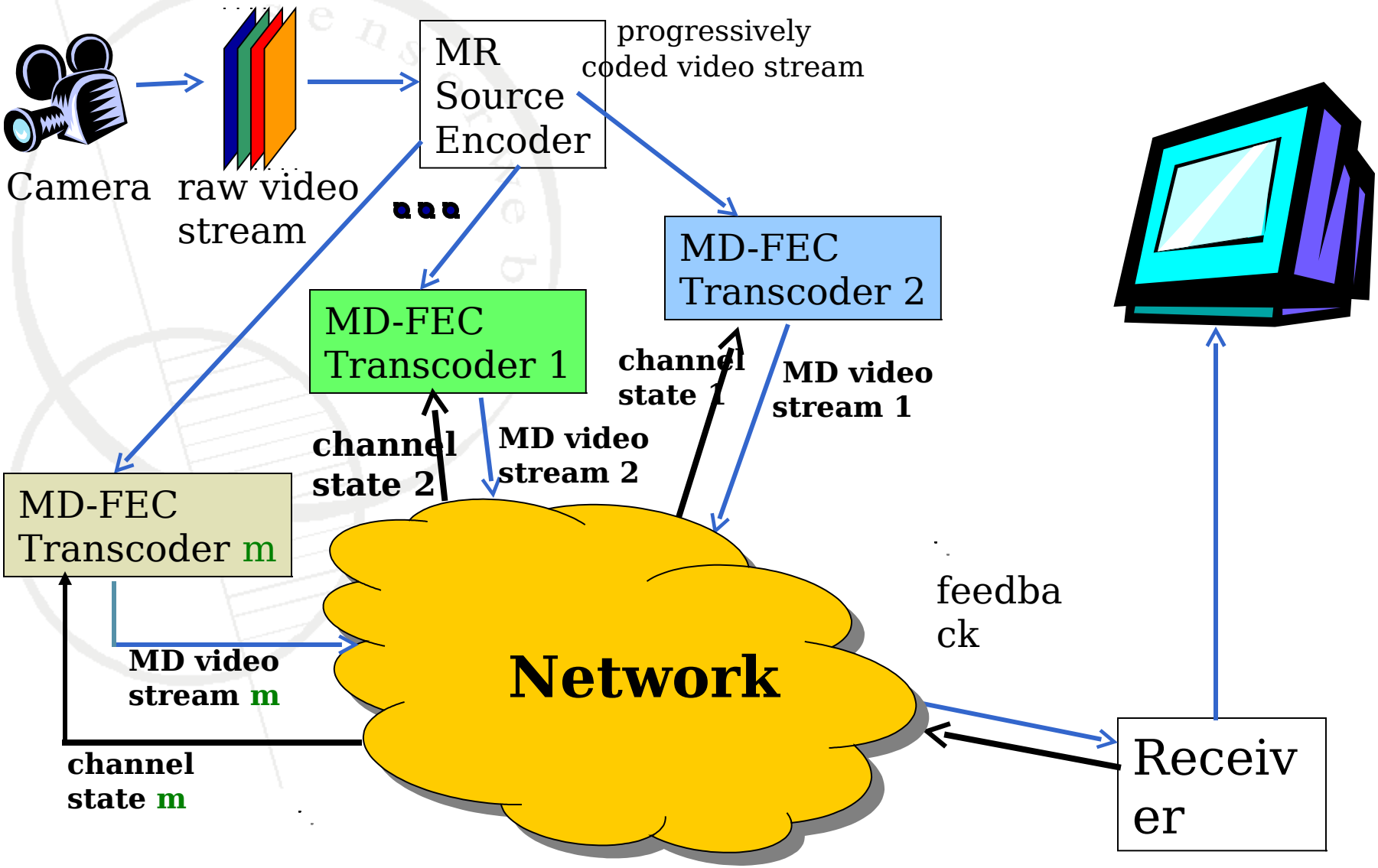


- MD-FEC (Multiple Descriptions through Forward Error Correction Codes) packet stream insensitive to ‘position’ of loss.
- MD-FEC rate markers can be optimized to dynamically adapt to **bo** instantaneous channel conditions and the source content. *(Puri & Ra*

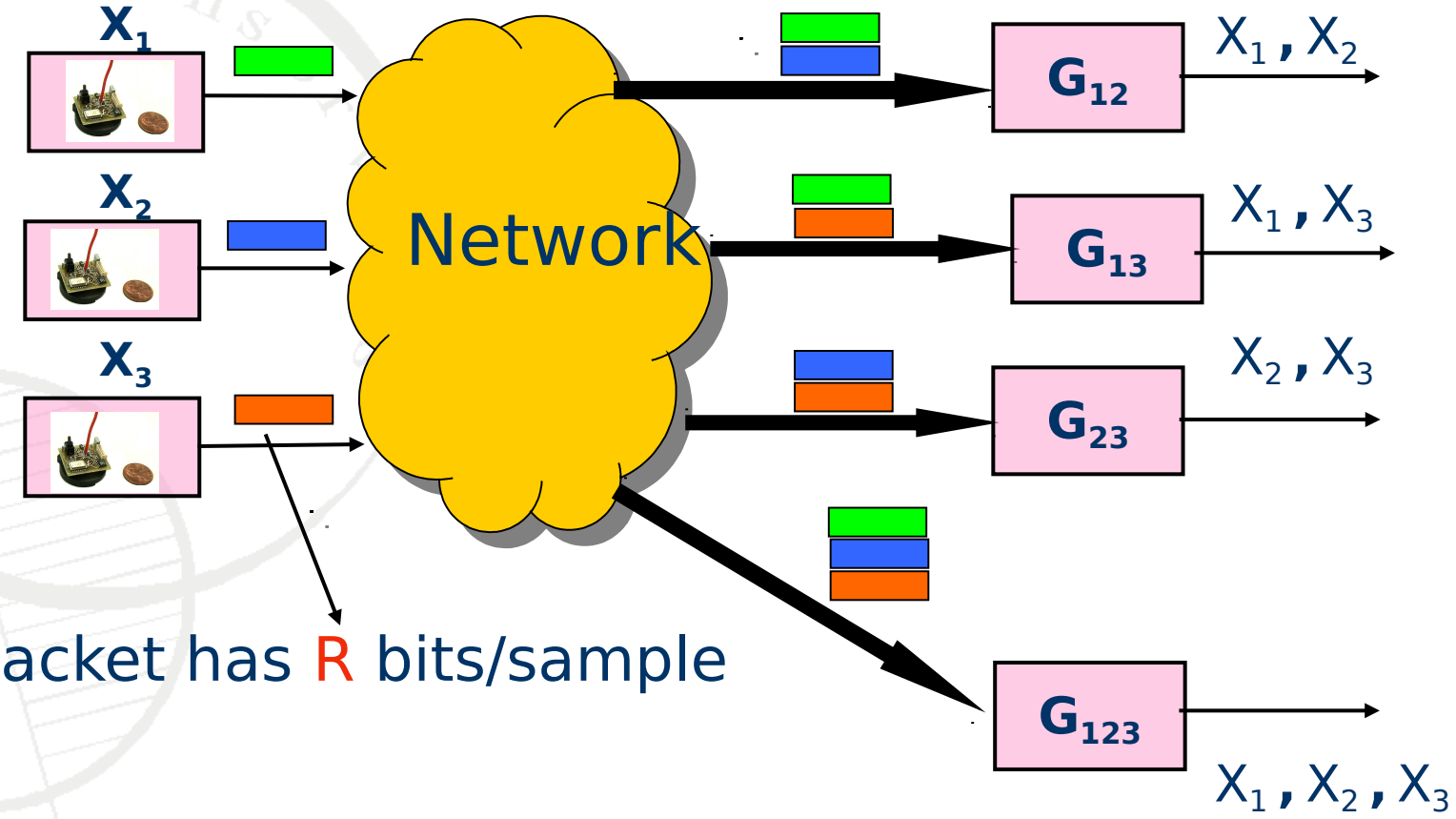
Outline of Solution (Single Receiver Case)

- Use a progressive bit stream to ensure graceful degradation.
- Find the loss rates and total bandwidth from each server to the client and calculate the “net” loss rate and bandwidth to the client.
- Apply the “MD-FEC” framework now that the problem is reduced to a point-to-point problem.

Distributed streaming paradigm: end-to-end system arch



“Robustified” distributed compression



Each packet has R bits/sample

Consider symmetric case: $\mathbf{H(X_i)=}H_1$, $\mathbf{H(X_i,X_j)=}H_2$, $\mathbf{H(X_i,X_j,X_k)=}H_3$:

- $R=H_3/3$ ➔ **Fully distributed**, maximally compressed, not robust t
- $R=H_2/2$ ➔ **Fully distributed**, minimally redundant, robust to ***any***

Future challenges:

- Integrate “distributed learning” aspect into framework
  - Extend to arbitrary correlation structures
- Incorporate accurate statistical sensor models:
  - Wavelet mixture models for audio-visual data
- Retain end-goal while optimizing system components:
  - **e.g.: estimation, detection, tracking, routing, transmission;**
  - **impose bandwidth/energy/computational constraints**
- Progress on network information theory & constructive algorithms
- Extend theory/algorithms for incorporating robustness/reliability
- Target specific application scenarios of interest.

1-D Vehicle tracking

For each vehicle, there are two parameters:

1. t_0 – the time the vehicle passes through point p = 0
2. v – the speed of the vehicle (assume constant velocity)

Node i at position p_i sees the vehicle at time t_i:

$t_i = t_0 + (1/v)p_i$

Combining all nodes, Ax = b with:

$$b = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \dots \\ t_n \end{bmatrix} \quad A = \begin{bmatrix} 1 & p_1 \\ 1 & p_2 \\ 1 & p_3 \\ \dots & \dots \\ 1 & p_n \end{bmatrix} \quad x = \begin{bmatrix} 1/v \\ t_0 \end{bmatrix}$$

$x = (A^T A)^{-1} A^T b$ Matrix inversion is only a 2x2!

Update Node Positions

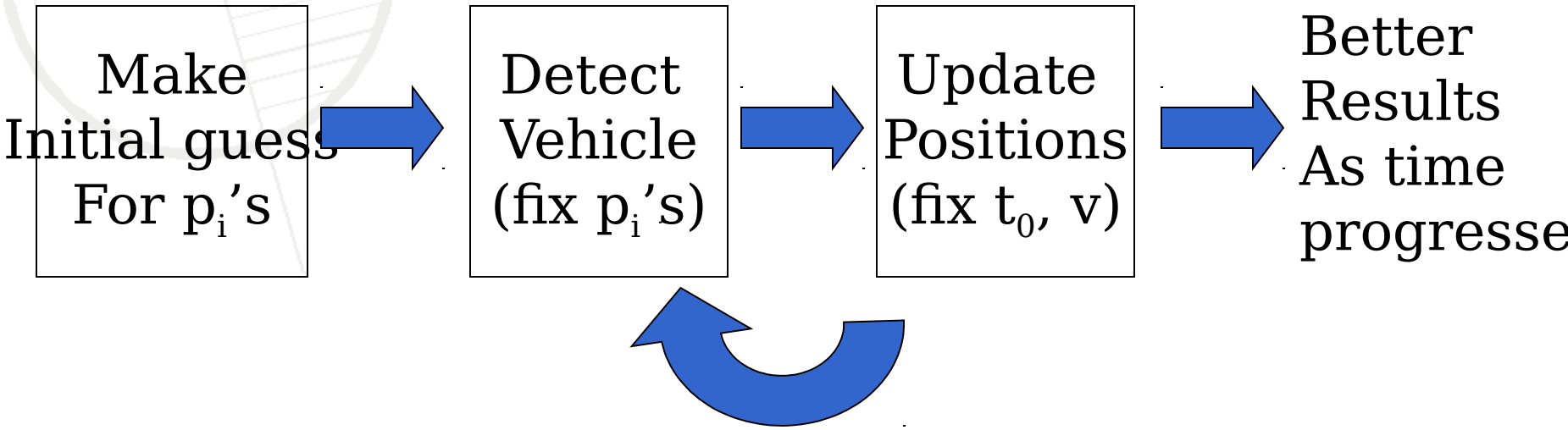
Once we calculate v, go back and make a new guess at each p_i

$t_i = (1/v)p_i^{new} + t_0$

$p_i^{new} = (t_i - t_0)v$

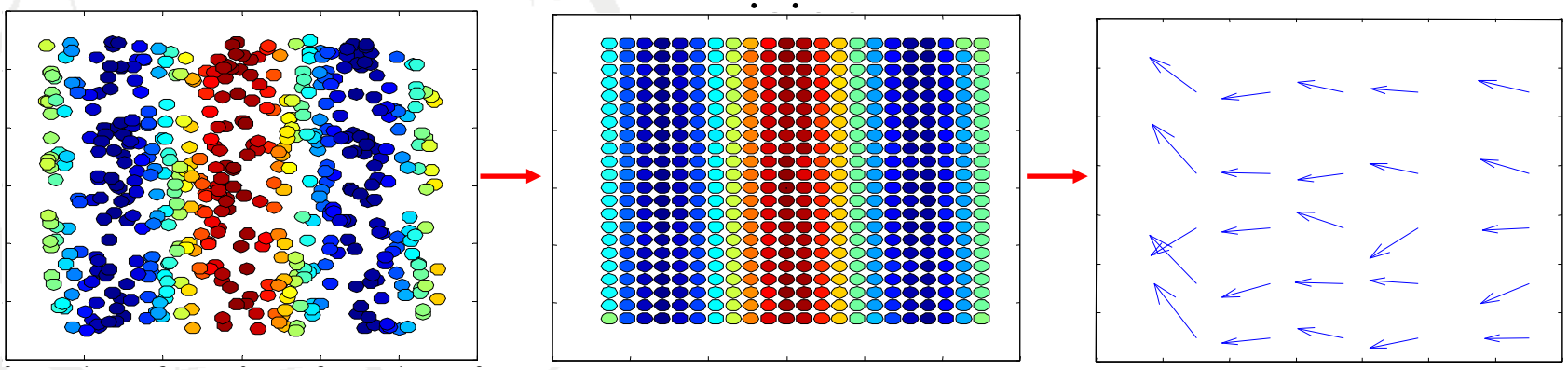
Update according to some non-catastrophic weighted rule like:

$$p_i^{next} = \frac{1}{eventnum} p_i^{new} + \frac{eventnum-1}{eventnum} p_i^{old}$$

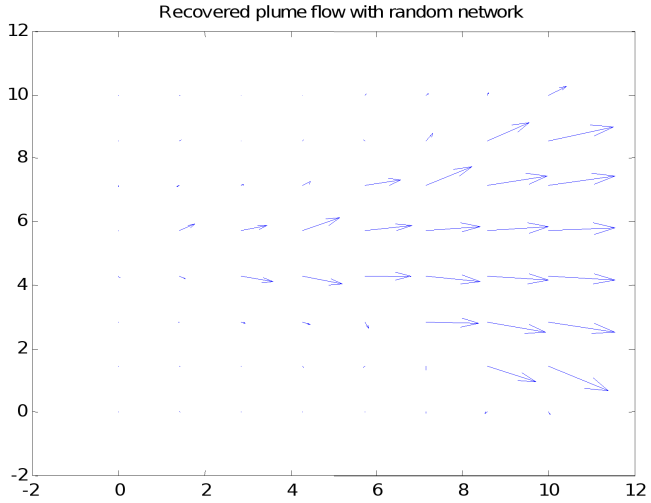


Dynamic Data Fusion

Use a node-pixel analogy to exploit algorithms from computer



Each sensor reading is akin to a pixel intensity at some (x,y) location.



By interpolating node positions to regular grid points, standard differentiation techniques are used to determine the direction of flow. This can be done in a distributed fashion.

Left: Chemical plume is tracked through